AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph that begins on page 9, line 3, with the following amended paragraph:

Further information on the background of protecting digital content can be found from the following three sources[[.]]: "*Music on the Internet and the Intellectual Property Protection Problem*" by Jack Lacy, James Snyder, David Maher, of AT&T Labs, Florham Park, NJ ~~available online URL http://www.a2bmusic.com/about/papers/musicipp.htm.~~ ; ~~Cryptographically~~ cryptographically protected container, called DigiBox, in the article "*Securing the Content, Not the Wire for Information Commerce*" by Olin Sibert, David Bernstein and David Van Wie, InterTrust Technologies Corp. Sunnyvale, CA ~~available online URL http://www.intertrust.com/architecture/stc.html. And~~; and "*Cryptolope Container Technology*", an IBM White Paper~~, available online URL http:///cyptolope.ibm.com/white.htm.~~ (refer to online URL www.microsoft.com for more information).

Please replace the paragraph that begins on page 9, line 10, with the following amended paragraph:

Yet[[, still]] another reason digital content providers have been slow to adopt electronic distribution is the ability for end-users to make unauthorized recordings of digital content once the content has been loaded on to an end-user device. This is especially problematic when the content is being played or rendered on an end-user device. Typically when the digital content is played, the content must be decrypted. During this instance when the digital content is decrypted, unauthorized digital copies are often times recorded. One method that can be used to make a digital copy is to open a recorder application, such as the Microsoft Media Recorder and record what is currently being played. Accordingly, a need exists to overcome this ~~shortcomings~~ shortcoming.

Please replace the paragraph that begins on page 9, line 18, with the following amended paragraph:

Continuing further, another shortcoming is that most multimedia [[end-user-systems]] end-user devices have the ability to play a multimedia file, and to record a multimedia file using audio output and audio input devices and/or ports and a patch cable. During the playing or rendering of digital content, the audio being played is patched from the output line of the sound card to the input line of the sound card. Accordingly, a need exists that will allow for encrypted digital content to be decrypted and played while blocking the recording of the content on the end-user device using an audio patch cable to wavein devices and/or ports, i.e., multimedia interfaces.

Please replace the paragraph that begins on page 9, line 25, with the following amended paragraph:

Still another shortcoming is that some multimedia systems have devices/applications that have the capability of capturing media streams but not for the purposes of recording the content, or these devices and/or ports are incapable of recording high quality content, and thus should not be confused with devices or applications attempting to capture [[the]] high quality streams being played. An example of such is a Voice Modem or Answering System application ~~which~~ that is capable of playing or recording low ~~bitrate~~ bit rate audio ~~being~~ received over a phone line. A need exists to discriminate and determine which devices/applications are "authorized" and "unauthorized" to be active during the content rendering process.

Please replace the paragraph that begins on page 10, line 9, with the following amended paragraph:

Briefly according to the present invention, there is provided a method, system and computer readable medium for the blocking of recording of digital Content at an [[end user]] end-user multimedia [[end-user-system]] device during the rendering of encrypted digital multimedia files. Before the process of rendering of encrypted digital multimedia can be started, all devices and/or ports capable of capturing the rendered media stream during playback are opened to ensure that this multimedia content is not recorded. This blocks the usage of the devices and/or ports that can be used to store un-encrypted content that has been decrypted for the purposes of playing or rendering. The method also includes an exception, which allows recording with permission ~~from the present invention~~.

Please replace the paragraph that begins on page 165, line 16, with the following amended paragraph:

FIG. 20 illustrates a high-level view of a multimedia End User Device(s) 109, and a block diagram of an audio feature card 2056 according to the present invention. With ~~this~~ the audio feature card 2056, an end-user has the ability to ~~encode or~~ record digital Content 113 such as music, even if the digital Content was encrypted. It is important to note that although the audio feature card 2056 is described as a feature card [[it]], the audio feature card 2056 may be built ~~in to~~ into the motherboard or split between the motherboard and the feature card. The audio feature card 2056 is communicatively coupled ~~with~~ to the system feature bus 2058, by the feature bus interface 2060. The audio feature card 2056 is designed to access [[the]] a storage media 2054 where the digital Content 113 is stored. The storage media 2054 can be a diskette, a network device, a disk drive, a CD, DVD, MiniDisk, DAT, cassette tape, laser disk or any other computer readable medium. In one embodiment, the digital Content 113 is encrypted so as to prevent ~~un authorized~~ unauthorized access. ~~Audio~~ The audio feature card 2056 is designed to convert [[the]] digital files to audio and back again. The audio feature card 2056 contains a special built-in processor, known as a DSP2062 (Digital Signal Processor) and control memory, which may be ROM 2064 (Read Only Memory), for processing digital audio Content 113 out through [[the]] speakers coupled to the End User Device(s) 109. Audio is a sequence of analog signals converted to digital signals during recording, using a microchip called an ADC (Analog to Digital Converter) 2068 on the audio feature card 2056. When audio, i.e., sound, is played, the digital signals are sent to the speakers where they are converted back to analog signals by a microchip called a DAC (Digital to Analog Converter) 2066 that generates the varied sound by converting a digital value to a given analog value as the Content 113 is read from the storage media 2054.

Please replace the paragraph that begins on page 166, line 9, with the following amended paragraph:

~~Digital~~ The digital Content 113 is usually compressed for storage and/or faster transmission. ~~Digital~~ The digital Content 113 is sent in short stand-alone segments. ~~One~~ An example is a Wave file format. In order for end-users to receive sound in real-time for a multimedia effect, for listening to music, or in order to take part in an audio or video conference, sound must be delivered as streaming sound. More advanced audio cards support wavetable, or precaptured tables of sound. A widely deployed format for Content 113 ~~format~~ is MPEG Layer3. The audio feature card 2056 is connected to a speaker 2072 by [[a]] an audio cable 2074 to [[the]] an audio out 2070. In order to simplify the drawings, only [[the]] a right side speaker is shown. It should be understood for stereo and surround sound systems, two or more speakers are necessary.

Please replace the paragraph that begins on page 166, line 17, with the following amended paragraph:

The digital Content 113 is read from the storage media 2054, decrypted and/or uncompressed if necessary, [[and]] converted to an analog signal by the DAC 2066, and played [[or]], i.e., rendered, through the speaker. It is during this time of rendering that the analog signal is "in the clear" [[or]], i.e., decrypted. To make an unauthorized perfect copy in the prior art, an end-user can install a patch cable 2076 to connect the audio out 2070 to [[the]] a line in 2078. The use of [[a]] the patch cable 2076 or software, such as operating system functions, ~~which provides~~ that provides the same function as [[a]] the patch cable, allows the Content 113 to be stored back to the storage media 2054 with no encryption. This decrypted Content 113, if captured in the digital format, is perfect. No analog noise, no background noise no conversion errors, the digital Content 113 is a perfect copy of the original. The audio feature card 2056 could entail a digital out interface or could provide a direct interface to the digital content prior to the DAC conversion. This type of recording removes the encryption and allows a perfect unauthorized copy to be played[[,]] or further copied, or sent or sold or swapped through an online service such as through Napster.com or Gnutella.com or Audiogallaxy.com, and, in general, it can be used in any way, as it is now not limited by [[the]] an encryption scheme. The present invention prevents this type of unauthorized re-recording while playing encrypted Content 113.

Please replace the paragraph that begins on page 167, line 6, with the following amended paragraph:

The term predetermined audio quality is a minimum level of quality ~~that is~~ deemed by the provider of the Content 113 to be usable ~~by the provider of the Content 113~~. For example, the Content Provider(s) 101 may deem that any audio quality that is less than CD quality ~~audio~~ for the subsequent unauthorized ~~re-recording~~ copy is not a concern. CD quality is two channels (stereo) of information ~~which~~ that has been converted to a digital file using 44,100 samples per sec.

Please replace the paragraph that begins on page 167, line 11, with the following amended paragraph:

A multimedia platform is designed to record audio [[or]] and to render audio of the Content 113 of the Digital Content Library 196 that is stored [[or]] in and read from the storage [[196]] media 2054 of the End User Device(s) 109. [[This includes]] A multimedia platform has the ability to [[input]] accept inputted audio and to convert to digital format [[the]] analog wave forms from sources such as a microphone, a musical instrument (such as a synthesizer), a MIDI (Musical Instrument Digital Interface) device, or from a direct connection to [[the]] an output of a waveout device ~~which~~ that is rendering audio content. This ability requires both the hardware, such as described in the audio feature card 2056 of FIG. 20 above, and device drivers that interface between the audio feature card and the application program. The [[ability to render]] rendering of the digital content results in one or more audio streams being delivered to one or more speakers 2072. The term "wavein" as used ~~in this invention~~ herein means [[the]] hardware combined with a device driver that records Content 113 based on specific quality. This is a common term used in the Microsoft Windows Multimedia programming environment.

Please replace the paragraph that begins on page 167, line 21, with the following amended paragraph:

A full-duplex sound card is designed to allow [[the]] recording and playback at the same time. It is this type of audio platform that can simultaneously render the multimedia content and record the multimedia content. The ability to patch the multimedia content output to the multimedia content input on an [[end-user-system]] end-user system is obviated by opening all [[of]] the wavein device drivers.

Please replace the paragraph that begins on page 168, line 6, with the following amended paragraph:

Turning now to FIG. 21, which is a flow diagram for rendering of encrypted files on the End User Device(s) 109 according to the present invention. The flow is entered at steps 2102 and 2104 when there is a need to render [[or]], i.e., play Content 113. All non-approved wavein devices and/or ports of a certain quality or predetermined audio level are opened at step 2200. This process is further explained in FIG. 22 below. ~~After completion of rendering at~~ At step 2110, ~~the present invention closes and the content is rendered. That is~~ the song or other multimedia Content 113 is played, or rendered. If the process of opening non-approved wavein devices and/or ports of a predetermined quality fails, ~~this~~ a message of such failure is sent to the Player Application 195 and the flow exits at step 2116 without rendering the ~~content~~ Content 113. [[At]] After completion of step 2110, the Content 113 ~~that~~ has ~~been~~ finished playing, and the method of the present invention closes the waveout device at step 2112. Once ~~this~~ the closing of the waveout device is completed, the wavein devices and/or ports are closed at step 2114, and the flow exits at step 2116.

Please replace the paragraph that begins on page 168, line 19, with the following amended paragraph:

FIG. 22 is a flow diagram 2200 illustrating the blocking of recording of Content 113 during the playing [[or]], i.e., rendering of the Content 113. This is accomplished by the present invention by opening, and thus blocking, all available recording devices and/or ports. The flow is entered at step 2202 when the ~~present invention~~ End User Device(s) 109 gets a list of N wavein devices and/or ports using the Microsoft Windows API call wavingetnumdevs() at step 2204. It is important to note that although the above embodiment ~~above~~ describes wavein for audio using a Microsoft Windows API, other recording devices and/or ports other than audio recording devices and/or ports are within the true scope and spirit of the present invention. Other recording devices and/or ports include picture, video, and other multimedia recordings. Moreover, the present invention can use any other type of I/O devices and/or ports used for recording under an operating system. Given this list of wavein devices and/or ports numbered from 0 through MAX, the present invention sets [[the]] N to 0 at step 2206. The device # 0 capability, or ability, to record at predetermined quality such as CD quality, is ascertained at step 2208. Optionally, if, at step 2210, it is determined that device #0 cannot accomplish CD quality recording the method of the present invention ~~at step 2210~~ does not ~~bother to~~ ascertain ~~if it~~ whether device #0 is open ~~or not~~, and step 2218 is entered. An example of ~~this~~ such a device is a modem. If device # 0 does support a predetermined quality such as CD quality, e.g., better or equal to two channels (stereo) at 44,100 samples per second, ~~at step 2210~~ the device is checked to see if it is open at step 2212. If it is not open, it is opened at step 2214. If it is open, at ~~step2212~~ step 2212, the device is checked to see if it has permission to be open ~~by the present invention~~ during the rendering of the decrypted media at step 2216. If it has permission, the ~~present invention~~ flow enters step 2218. If it is open with no permission, a failure to open error is presented to the Player Application 195 at step 2220.

Please replace the paragraph that begins on page 169, line 17, with the following amended paragraph:

This completes the description of the method of the present invention during the playing, i.e., rendering of encrypted Content 113. All available recording devices and/or ports of a predetermined quality are opened, and, therefore, cannot be used for recording.

Please replace the paragraph that begins on page 169, line 22, with the following amended paragraph:

Turning now to FIG. 23, which is a flow diagram 2300 describing the requirement to record Content 113. The flow is entered at step 2302 with the authorization to record the Content 113 at step 2304. Any recording application for [[a]] predetermined quality files will search for a not open, i.e., available, wavein device. If encrypted media is being rendered at step 2306 and no wavein devices and/or ports of a predetermined quality are available at step 2308, [[and]] the flow exits at step 2310. If no encrypted media is being rendered at step 2306, the recording application can open a wavein device and record media at step 2312. The present invention does not interfere with ~~record~~ recording operations ~~while~~ when the Player Application 195 ~~does~~ is not actively rendering secure Content 113. Furthermore, failure to render, due to open wavein devices and/or ports being detected in flow diagram 2200, can be rectified by end-users closing all applications ~~which~~ that have an open wavein port, and retrying the rendering process.